International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 4, Issue 11, pp 20-25, November-2020

https://doi.org/10.47001/IRJIET/2020.411003

Implemented Text Rank based Automatic Text Summarization using Keyword Extraction

[1]Anurag Kumar Yadav, [2]Mukesh Kumar, [3]Ayonija Pathre

[1]Research Scholar, Department of CSE, RNTU, India
[2]HOD, Assistant Professor, Department of CSE, RNTU, India
[3]Assistant Professor, Department of CSE, RNTU, India

***Abstract* - The Automatic text summarization is a kind of technique for generating a precise and concise summary. For the summarization, the machine learning algorithm can be trained to comprehend documents and identify the sections that communicate important facts and information before producing the required summarized texts. In this research paper, we are adopting implemented text rank based Automatic Text Summarization to achieve quality summary and quality keywords which are required for text summarization. The performance of the proposed summarization system is measured in terms of F-measure score. The outcome of research study shows around 2 percentage better outcomes as compared to earlier work.**

Keywords: Automatic text summarization, Keyword Extraction, F-measure, Machine Learning.

I. INTRODUCTION

The Automatic text summarization is to transform lengthy documents into short version something which can be difficult and costly to undertake if done manually.

For the summarization, the machine learning algorithm can be trained to comprehend documents and identify the sections that communicate important facts and information before producing the required summarized texts. For an example, the image of the news article has been fed into a machine learning algorithm to produce a summary [1].

In a specific way, there are two different approaches for text automatic summarization that are (a) extraction and (b) abstraction. The extractive summarization methods work by identifying important sections of the text and generating them verbatim; so they depend only on extraction of sentences from the original text while abstractive summarization methods target at producing important material in a new way.

This can be explained as they interpret and examine the text using advanced natural language techniques in order to generate a new shorter text that conveys the most critical information from the original text [2].

II. THE NEED FOR TEXT SUMMARIZATION

The summarization by manual generating a summary can be time consuming and wearisome. The automatic text summarization promises to conquer such difficulties and allow you to generate the key ideas in a piece of writing easily. With the current explosion of data circulating the digital space, that is mostly non-structured textual data. So, there is a need to develop automatic text summarization tools which allow people to get insights from them easily. Presently, it can be enjoy quick access to enormous amounts of information. The implementation of summarization can enhance the readability of documents, reduce the time spent in researching for information that allow for more information to be fitted in a particular area [3-5].

III. LITERATURE REVIEW

The surveys of some of the most powerful existing Automatic Text Summarization techniques have been studied. The author [6] discussed the MAP algorithm which is the platform for multi-document text summarization. This paper provides the functionality of the MAP, a public domain, open source and comprehensive, multi-document multilingual summarization. It is widely used by more than 580 companies and organizations. It is applied in various applications ranging from the mobile phone technology to web page for summarization and also for novelty identification. The author [7] combined n- gram and dependency word pair for multi-document summarization task.

The author [8] proposed an evolutionary optimization algorithm for multi-document summarization system. This system creates a summary by collecting the salient sentence from the multiple documents. So this approach takes the benefit of the summary to document collection, and sentence-to-sentence collection and sentence-to-document collection to choose the most important sentences from the multiple documents. Therefore, it reduces the redundancy in the document summary.

The author [9] proposed a weighted voting summation of Self Organization Map like ensembles. The weighted voting

© 2017-2020 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 4, Issue 11, pp 20-25, November-2020

https://doi.org/10.47001/IRJIET/2020.411003

superposition was used for the outcomes of an ensemble of Self Organization Map. The purpose of this algorithm was to attain the minimal topographic error in the map. So, a weighted voting process was done among the neurons to calculate the properties of the neurons located in the resulting map.

The author in [10] commenced an approach for multi-document summarization using Latent Semantic Analysis. All among the existing multi-document summarization approaches, the Latent Semantic Analysis was a unique concept, which uses the latent semantic information rather than the original features. It has chosen the sentence individually to remove the redundant sentences. The author [11] has discussed an extractive document summarization.

IV. TEXT SUMMARIZATION OF A WIKI NEWS ARTICLE

Text summarization can be applied on web articles for checking its efficiency. To keep things simple, from Python NLTK toolkit, any machine learning library can be used [12-14].

Here are the steps for creating a simple text summarizer by using Python programming.

Step 1: Preparation of the data

Here, we want to summarize the information found on Wiki news article, which gives an overview of the main happenings during the current era.

We begin by importing the essential libraries for fetching data from the web page. The Beautiful Soup library is used for parsing the page while the url library is used for connecting to the page and retrieving the HTML.

Step 2: Processing the data

To ensure the scrapped textual data is as noise-free as possible, we perform some basic text cleaning. To assist us to do the processing, we import a list of stop words from the NLTK library.

If, in any case, the word was previously available in the dictionary, its value is updated by the value 1. Otherwise, if the word is recognized for the first time, its value is set to 1.

For example, the frequency table should looks like the following:

WORD	FREQUENCY
century	7
world	4
United States	3
computer	1

Step 3: Tokenizing the article into sentences

To split the article content into a set of sentences, we use the built-in method from the Python NLTK library.

Step 4: Finding the weighted frequencies of the sentences

For the evaluation, the score for every sentence in the text, we analyze the frequency of occurrence of each term. In this condition, we score each sentence by its words; which are adding the frequency of each important word found in the sentence.

Step 5: Calculating the threshold of the sentences

To further pinch the kind of sentences eligible for summarization, we create the average score for the sentences. With this threshold limit, we avoid selecting the sentences with a lower score than the average score.

Step 6: Gathering the summary

At the end, since we have all the required parameters, we then generate a summary for the article.

4.1 Text Rank based Algorithm

The algorithm Text Rank is a graph-based ranking algorithm for text processing which is used in order to find the most relevant sentences in text and also to find keywords [15]. Some of the instructions for this algorithm is given below:

- In place of web pages, we use sentences.
- Similarity between any two sentences is used as an equivalent to the web page transition probability.
- The similarity scores are stored in a square matrix, similar to the matrix M used for Page Rank

4.2 Page Rank based Algorithm

The Text Rank is a kind of algorithm which is based on Page Rank method that often used in keyword extraction and text summarization. In this research study, we will help you to understand how Page Rank works with a keyword extraction with example and show the implementation using Python language [16].

© 2017-2020 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 4, Issue 11, pp 20-25, November-2020

https://doi.org/10.47001/IRJIET/2020.411003

The Page Rank is used to calculate the weight for the web pages. We take all web pages as a large directed graph. In this graph, a node is treated as a webpage. If webpage 'A' has the link to web page 'B', it is represented as a directed edge from 'A' to 'B'.

After all construct the whole graph, and we can assign weights for web pages by the following formula.

$$S(Vi) = (1-d) + d * \sum_{x=1}^{j} \frac{1}{Out(Vj)} S(Vj)$$

Where

S(Vi) = the weight of website i
d = damping factor, in case of no outgoing links
In(Vi) – inbound links of I, which is a set
Out(Vj) – outgoing links of j, which is a set

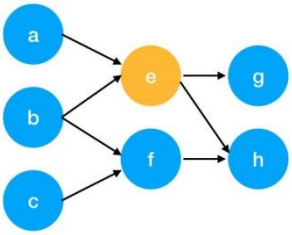

Figure 1: Demonstration of web page link representation using graph

We rewrite the summation part in the above function to a simpler version. We can get the weight of webpage e by the following function:

$$S(Ve) = (1-d) + d * (S(Va) + \frac{1}{2} S(Vb)$$

Here we watch the weight of the webpage e is dependent on the weights of its inbound pages. We need to run this iteration much time to get the final weight. In the initialization, the importance of each webpage is 1.

The Page Rank algorithm is used basically for ranking web pages in online search results. For example, suppose we have four web pages namely; w1, w2, w3, and w4. These pages contain links pointing to one another. In which some pages might have no link; these are called dangling pages.

webpage	links
w1	[w4, w2]
w2	[w3, w1]
w3	[ ]
w4	[w1]

- Web page w1 has links directing to w2 and w4
- w2 has links for w3 and w1
- w4 has links only for the web page w1
- w3 has no links and hence it will be called a dangling page

To obtain the rank of these pages, we will have to compute a score called the Page Rank score. This score is the probability of a user visiting that page.

To capture the probabilities of users navigating from one page to another, we just create a square matrix M, having n rows and n columns, where n is the number of web pages.

The initialization of the probabilities is explained in the steps shown below:

1. The probability of reaching from page i to j, which mean M[i][j], is initialized with "1/(no. of unique links in web page wi)"
2. If there is no link between the page i and j, then the probability will be initialized with "0"
3. If a user has landed on a dangling page, then it is assumed that he is equally likely to transition to any page. Therefore, M[i][j] will be initialized with "1/(number of web pages)"

So, in this case, the matrix M will be initialized as given below:

M =

	w1	w2	w3	w4
w1	0	0.5	0	0.5
w2	0.5	0	0.5	0
w3	0.25	0.25	0.25	0.25
w4	1	0	0	0

At the end, the values in this matrix will be updated in an iterative fashion to arrive at the web page rankings.

V. PROPOSED METHODOLOGY

5.1 Implementation in Page Rank Algorithm

In implementation we use a matrix to represent the inbound and outbound links among a, b, e, f in the graph shown in Figure 1.

	a	b	e	F
a	0	0	0	0
b	0	0	0	0
e	1	1	0	0
f	0	1	0	0

The each node in the row means the inbound links from other nodes. For an example, for the e row, node a and b have outbound links to node e. It will simplify the calculation for updating the weight According to the function; we need to normalize each column.

© 2017-2020 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 4, Issue 11, pp 20-25, November-2020

https://doi.org/10.47001/IRJIET/2020.411003

	a	b	e	f
a	0	0	0	0
b	0	0	0	0
e	1	0.5	0	0
f	0	0.5	0	0

We use this matrix to multiply with the weight of all nodes.

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1.5 \\ 0.5 \end{bmatrix}$$

Here if we change the directed edge, as the undirected edge, we can change the matrix correspondingly.

	a	b	e	F
a	0	0	1	0
b	0	0	1	1
e	1	1	0	0
f	0	1	0	0

Now normalize the above data as follows:

	a	b	e	F
a	0	0	0.5	0
b	0	0	0.5	1
e	1	0.5	0	0
f	0	0.5	0	0

5.2 Keyword Extraction by Implemented TextRank

The Page Rank algorithm is used for webpage ranking and Text Rank is used for text ranking. The webpage in Page Rank is the text in TextRank, hence the basic idea is the same [17].

To understand the perception of keyword extraction, we just split a document into several sentences, and only store those words with specific POS tags. We use spa Cy function for POS (Part of Speech) tagging.

```
content = '''The Wandering Earth, described as China first big-budget science fiction thriller, quietly made it onto screens at AMC theaters in North America this weekend, and it shows a new side of Chinese filmmaking — one focused toward futuristic spectacles rather than China's traditionally grand, massive historical epics. At the same time, The Wandering Earth feels like a throwback to a few familiar eras of American filmmaking.
```

Here split the above paragraph into three sentences.

```
"The Wandering Earth, described as China first big-budget science fiction thriller, quietly made it onto screens at AMC theaters in North America this weekend, and it shows a new side of Chinese filmmaking – one focused toward futuristic spectacles rather than China's traditionally grand, massive historical epics. At the same time, The Wandering Earth feels like a throwback to a few familiar eras of American filmmaking.
```

In the above text, most of the words in the sentence are not useful to determine the usefulness, so we only consider the words with NOUN, PRONOUN, and VERB by using POS tags. It is noticed that this is optional; we can also use all words.

In the above text, each word is a node in Page Rank. So we set the window size as k.

W1, W2, W3, W4, W5, …….. Wn

Here, [W1, W2, ……., Wk], [W2, W3, ……., W{k+1}], [W3, W4, ……., W{k+2}] are windows.

Any two-word pairs in a window are considered have an undirected edge.

We take

```
[time, Wandering, Earth, feels, throwback, eras, filmmaking]
```

As the example, and set the window size as k = 4,

Now, we get 4 windows,

```
[time, Wandering, Earth, feels], [Wandering, Earth, feels, throwback], [Earth, feels, throwback, eras], [feels, throwback, eras, filmmaking].
```

For window

```
[time, Wandering, Earth, feels],
```

Any two words pair has an undirected edge. So we get

```
(time, Wandering), (time, Earth), (time, feels), (Wandering, Earth), (Wandering, feels), (Earth, feels).
```

Based on this graph, we can calculate the weight for each node i.e. each word. The most important words can be used as

© 2017-2020 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 4, Issue 11, pp 20-25, November-2020
https://doi.org/10.47001/IRJIET/2020.411003

keywords. To check the efficiency of the implemented Text Rank Algorithm, we calculate Precision, Recall and F-measure value of the dataset.

The performance of the proposed summarization system is measured in terms of F-measure score with respect to the ROUGE-2 metrics. On the basis of Table 1, we can calculate the best F-score of the summaries.

Table 1: Comparison of proposed implemented Text Rank algorithm with existing method

Metrics	Implemented Text Rank algorithm	Existing Text Rank algorithm
Precision	86.4	80.2
Recall	83.2	78.7
F-measure	93.0	91.0

The above Table shows the comparison between the implemented Text Rank algorithm and the existing Text Rank method. From the results, it is clear that the proposed method works better than the existing method.

VI. CONCLUSIONS

The increasing progression of the Internet has made a huge amount of information available. It is very difficult for humans to summarize huge amounts of text. So, there is an immense need for automatic summarization systems in this age of information excess. Due to the rapid growth of knowledge and use of Internet, there is information overload. This problem can be solved, if there are robust text summarizers which produces a summary of document to help user. Hence, there is a necessity to develop system where a user can efficiently retrieve and get a summarized document. One potential solution is to summarize a document using either extractive or abstractive methods. The text summarization by extractive is easier to build.

In this research work, we emphasized various extractive approaches for single and multi-document summarization. We have discussed some of the most extensively used methods and machine learning approach. It delivers a good insight into recent trends and progresses in automatic summarization methods and describes the up-to-the-minute in this research area.

REFERENCES

[1] R.Mihalfcea, and P.Tasrau, "TextRank: Bringing Order into Texts." *In Proceedings of Empirical Methods in Natural Language Processing (EMNLP).* pp. 404-411. 2004.

[2] Z. Pei-yhing, and L. Crun-he, "Automatic Text Summarization based on Sentences Clustering and Extraction," *Proceeding of the 2nd IEEE International Conference on Computer Science and Information Technology*. pp. 167-170. 2009.

[3] D. Bleei, A. Ng, and M. Jssordan "Latent Dirichlet allocation". *In Journal of Machine Learning Research*, 3:993–1022, January 2003.

[4] Barzilay, R. and Elhadad, M. "Using lexical chains for text summarization." *in Proceedings ISTS'97.* pg. 38-41 (1997).

[5] Radev, D. R. and McKeown, K. "Generating natural language summaries from multiple on-line sources", *Computational Linguistics*, 24(3):469–500 (2002).

[6] S. Baherjee, P. Mittra and K. Suggiyama "Multi-Document Abstractive Summarization Using ILP Based Multi-Sentence Compression", *in Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015).*

[7] Mc Keeown, K.R. and Radev, Dared R. (1995). "Generating summaries of multiple news articles." *in Proceedings of SIGIR*, Seattle, Washington, pages 74–82, 1995.

[8] Jagadeesh J, Prasad Pingali, Vasudeva Varma "Sentence Extraction Based Single Document Summarization" *Workshop on Document Summarization*, IIIT Allahabad, 19th and 20th March, 2005.

[9] Kamala Suarkar, "Sentence Clustering-based Summarization of Multiple Text Documents", *TECHNIA – International Journal of Computing Science and Communication Technologies*, vol. 2, no. 1, Jul. 2009.

[10] F. Canaen Pemsbe and Tunga Günragör, "Automated Query-biased and Structure preserving Text Summarization on Web Documents", *in Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications*, İstanbul, June 2017.

[11] Reev Lawsrence H., Han Hyhoil, Naori Saya V., Yang Jonathan C., Schwimmer Tamara A., Brooks Ari D., "Concept Frequency Distribution in Biomedical Text Summarization", *ACM 15th Conference on Information and Knowledge Management (CIKM), Arlington*, VA, USA, 2016.

[12] Khaen Atif, Salm Nauomie, "A review on abstractive summarization Methods", *Journal of Theoretical and Applied Information Technology*, Vol. 59, 2014.

[13] Evas, Darak. K. "Similarity-based multilingual multi-document summarization", *Technical Report CUCS-014-05*, Columbia University (2012).

© 2017-2020 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 4, Issue 11, pp 20-25, November-2020
https://doi.org/10.47001/IRJIET/2020.411003

[14] A Edeundson, H. P., "New methods in automatic extracting." *Journal of the ACM,* 16(2):264–285, 1969.

[15] Marttins, Cailla Barandel and Luia Helena Machado Rino. "Revisiting UNLSumm: Improvement through a Case Study." (2002).

[16] A.Fattah and F. Ren, "Automatic text summarization," *World Academy of Science, Engineering and Technology,* vol. 37, p. 2008, 2008.

[17] "About wordnet", https://wordnet.princeton.edu/, accessed: 2015-05-15.

Citation of this Article:

Anurag Kumar Yadav, Mukesh Kumar, Ayonija Pathre, "Implemented Text Rank based Automatic Text Summarization using Keyword Extraction" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 4, Issue 11, pp 20-25, November 2020. https://doi.org/10.47001/IRJIET/2020.411003

****

© 2017-2020 IRJIET All Rights Reserved